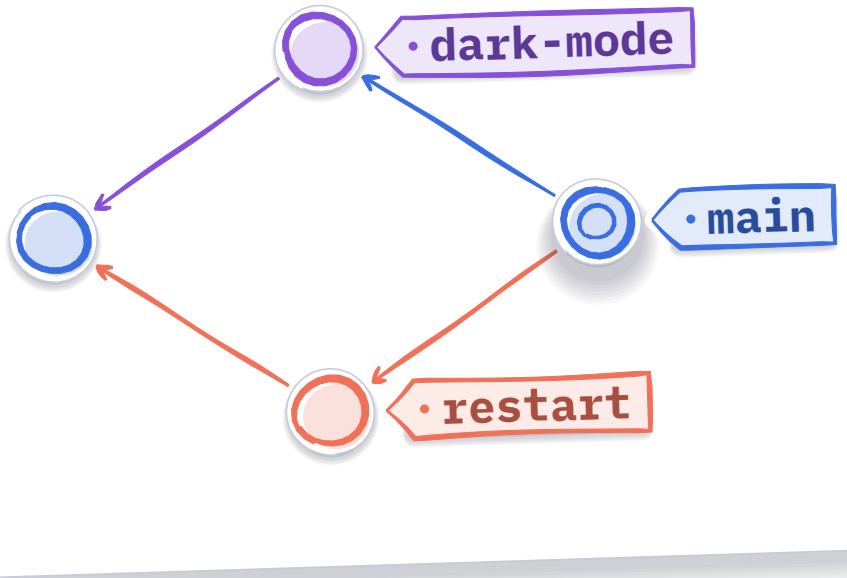
Prof. Aviv Zohar

Semester A, 2026–7

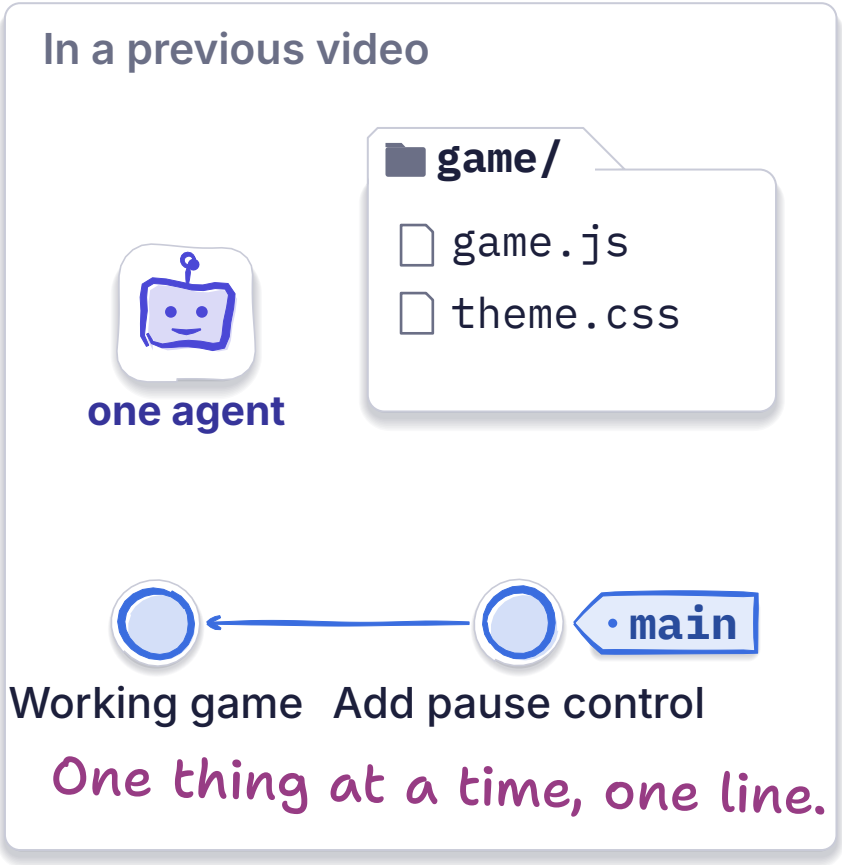
Separate features, then merge them

A branch and a worktree per agent, a fast-forward, and one conflict



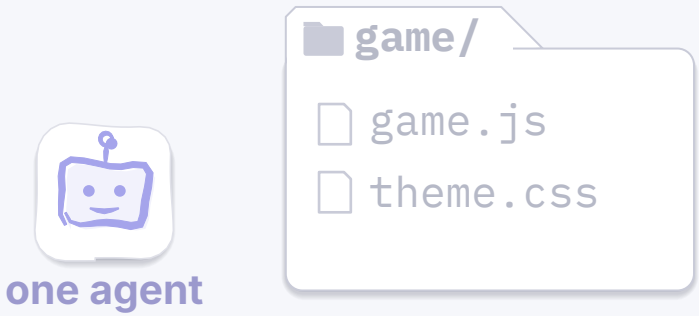
Two agents at once need three things

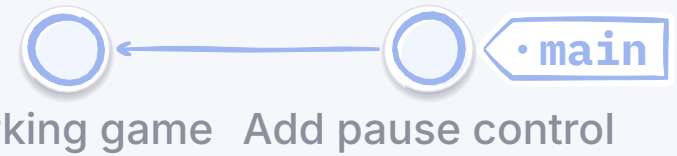
In a previous video



Two agents at once need three things

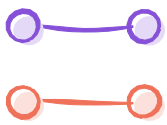
In a previous video







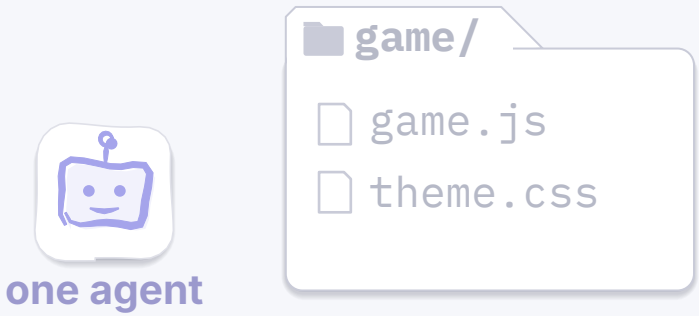
Its own files to edit
so half-finished changes stay apart



Its own saved work
so we review one feature without the other

Two agents at once need three things

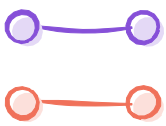
In a previous video





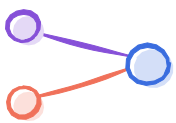
Its own files to edit `worktree`

so half-finished changes stay apart



Its own saved work `branch`

so we review one feature without the other



Brought together in `main` `merge`

the work we accept

Two agents, one folder



Two agents, one folder



Restart is gone, and nobody is told.

The last write wins.

Two agents, one folder



ⓐ a dark mode B never asked for
ⓑ a key that calls nothing

Separate tasks do not give the agents separate files.

A folder and a branch for each agent

Worktree: a separate working folder of the same repository.

game/



you

game.js

theme.css

game-dark/



agent A

game.js

theme.css

game-restart/



agent B

game.js

theme.css

one repository: the saved history all three folders share



A folder and a branch for each agent

Branch: a name for a line of committed work.

game/

you

- game.js
- theme.css

main

game-dark/

agent A

- game.js
- theme.css

dark-mode

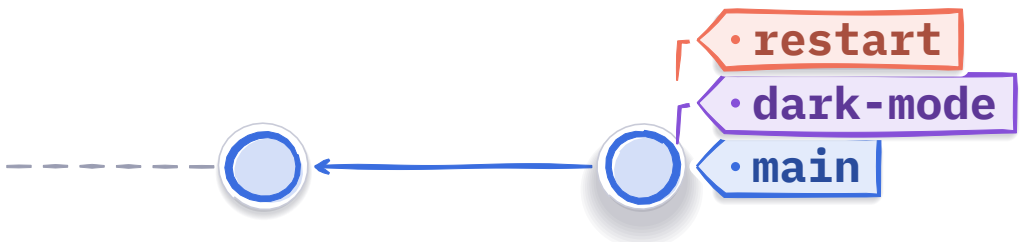
game-restart/

agent B

- game.js
- theme.css

restart

one repository: the saved history all three folders share



All three names point at the pause commit.

A folder and a branch for each agent

A branch is a name, not a copy of the history.

Separate in each folder: working files and staging area.

game/

you

game.js
theme.css

·main

game-dark/

agent A

game.js
theme.css

·dark-mode

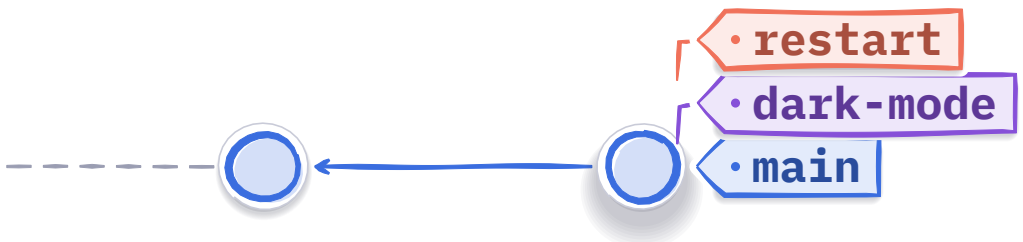
game-restart/

agent B

game.js
theme.css

·restart

one repository: the saved history all three folders share



Shared: one repository.

A folder and a branch for each agent

Its own branch and its own folder.

game/

you

game.js
theme.css

main

game-dark/

agent A

game.js
theme.css

dark-mode

game-restart/

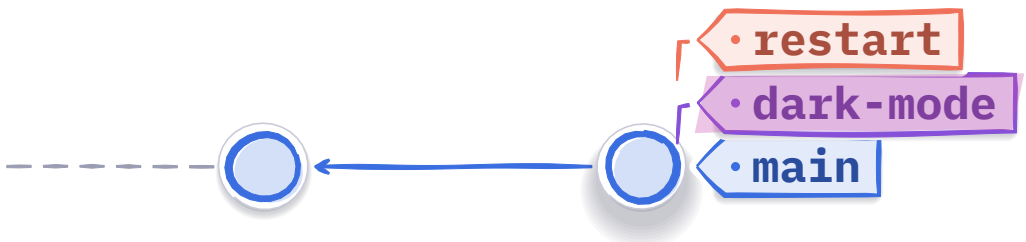
agent B

game.js
theme.css

restart

The folder: where it reads and edits files.

one repository: the saved history all three folders share



The branch: the name for the commits it saves.

Ask for the branch and the folder first

agent session ~/game

you typed by you

Create a dark-mode branch from main in its own worktree, and work there.

tool call asked by the model

```
Bash git worktree add -b dark-mode ../game-dark main
```



no game-dark/ yet

Ask for the branch and the folder first

agent session ~/game

you typed by you

Create a dark-mode branch from main in its own worktree, and work there.

tool call asked by the model

Bash `git worktree add -b dark-mode ../game-dark main`



no game-dark/ yet

```
git worktree add -b dark-mode ../game-dark main
                 ⓐ            ⓑ           ⓒ
```

ⓐ creates the branch

ⓑ names the new folder, next to game/

ⓒ the commit it starts from

Ask for the branch and the folder first

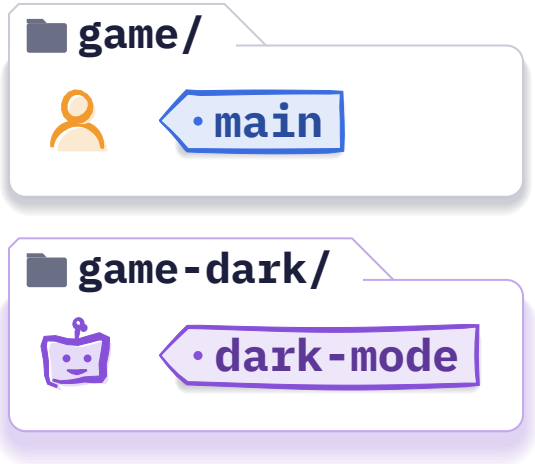
agent session ~/game

you typed by you

Create a dark-mode branch from main in its own worktree, and work there.

tool call asked by the model

```
Bash git worktree add -b dark-mode ../game-dark main
```



Its files are checked out here, ready to edit.

Ask for the branch and the folder first

agent session ~/game

you typed by you

Create a dark-mode branch from main in its own worktree, and work there.

tool call asked by the model

Bash `git worktree add -b dark-mode ../game-dark main`

tool result run by the harness

```
Preparing worktree (new branch 'dark-mode')
HEAD is now at 686e1c5 Add pause control
```



The agent works in `game-dark/`.

It starts at the pause commit, where `main` points.

Ask for the branch and the folder first

agent session ~/game

you typed by you

Create a dark-mode branch from main in its own worktree, and work there.

tool call asked by the model

Bash `git worktree add -b dark-mode ../game-dark main`

tool result run by the harness

```
Preparing worktree (new branch 'dark-mode')
HEAD is now at 686e1c5 Add pause control
```

you typed by you

Do the same for restart.

tool call asked by the model

Bash `git worktree add -b restart ../game-restart main`







Both start from the pause commit. game/ stays as it was.

A commit moves its branch's name

agent session ~/game-dark

tool call agent A

Bash

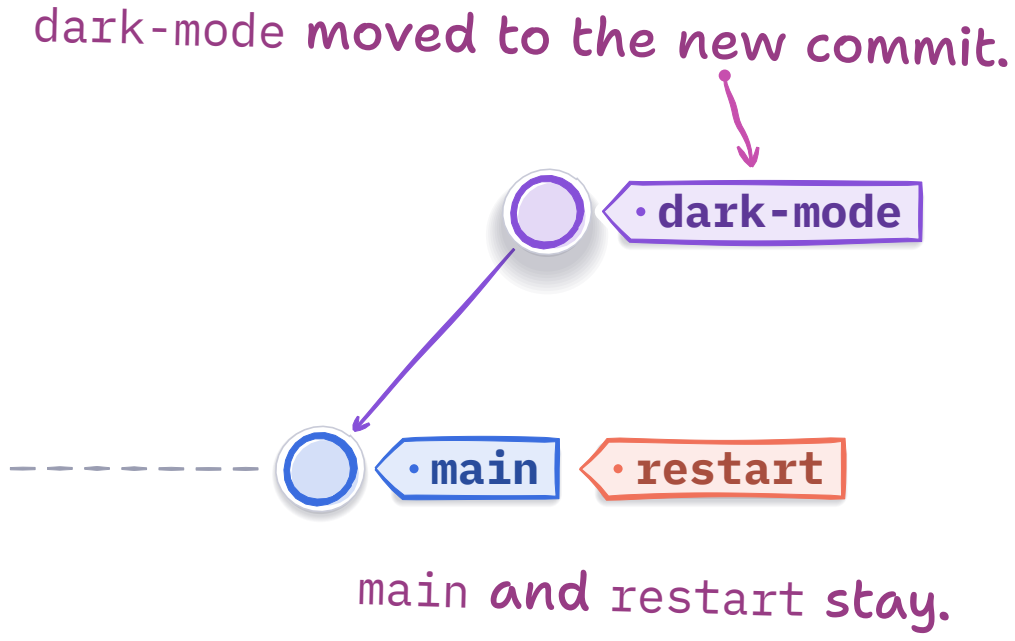
```
git add game.js theme.css
git commit -m "Add dark mode"
```

agent session ~/game-restart

tool call agent B

Bash

```
git add game.js
git commit -m "Add restart control"
```



A commit moves its branch's name

agent session ~/game-dark

tool call agent A

Bash

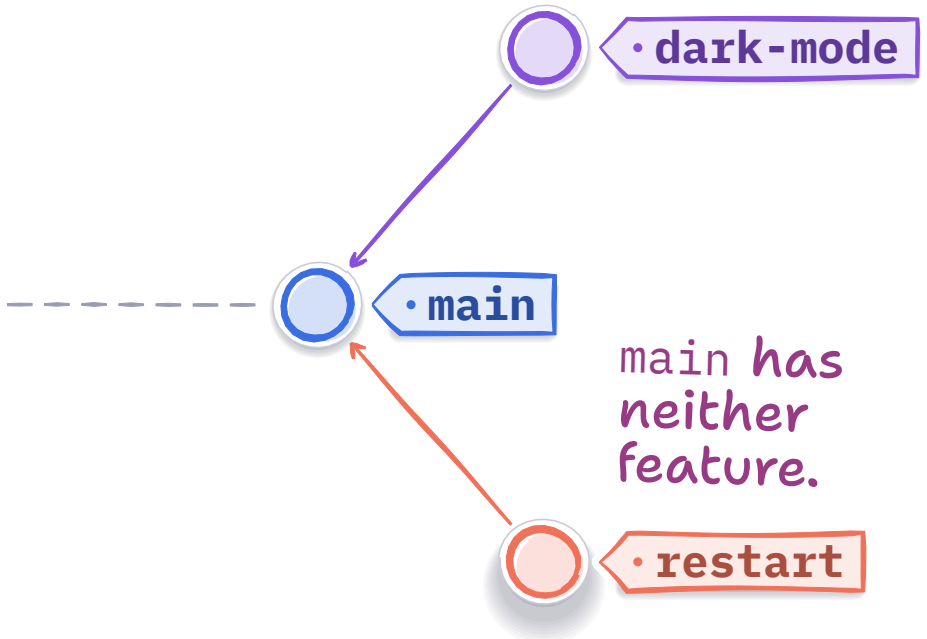
```
git add game.js theme.css
git commit -m "Add dark mode"
```

agent session ~/game-restart

tool call agent B

Bash

```
git add game.js
git commit -m "Add restart control"
```



restart moved to its new commit.

A commit moves its branch's name

agent session ~/game-dark

tool call agent A

Bash

```
git add game.js theme.css
git commit -m "Add dark mode"
```

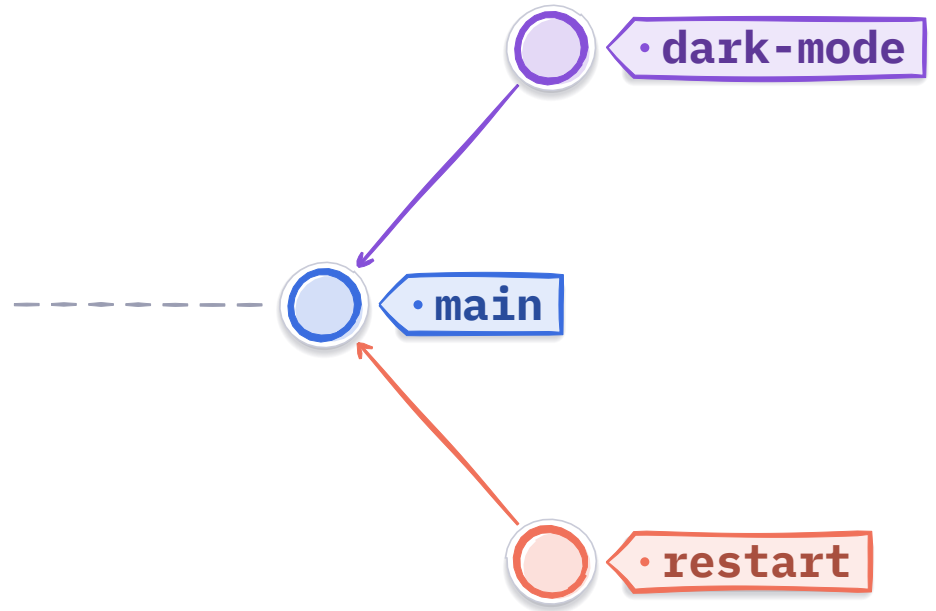
agent session ~/game-restart

tool call agent B

Bash

```
git add game.js
git commit -m "Add restart control"
```

One history, seen from every folder.



The working files stay separate: game/ still has no dark mode.

Review a feature before merging it

shell

```
~/game-dark $ git diff main
```

you type this one

2 files changed

- game.js + the game logic
- theme.css + the colours

game.js — an excerpt of the diff

```
 7  const STEP = 6;
 8
-   const controlsHelp = "P: pause";
+   const controlsHelp = "P: pause | D: dark mode";
10
11  const INVADER = [
```

The controls text gains the dark-mode key.

Review a feature before merging it

shell

```
~/game-dark $ git diff main
```

you type this one

2 files changed

- `game.js` + the game logic
- `theme.css` + the colours

game.js — an excerpt of the diff

```
 7  const STEP = 6;
 8
-   const controlsHelp = "P: pause";
+   const controlsHelp = "P: pause | D: dark mode";
10
11  const INVADER = [
```



screenshot · dark mode, run in `game-dark/`

- The colours change
- `D` toggles the theme
- `P` still pauses

Ready to merge when the changes make sense and it does what we asked.

Merge dark mode into main

agent session ~/game

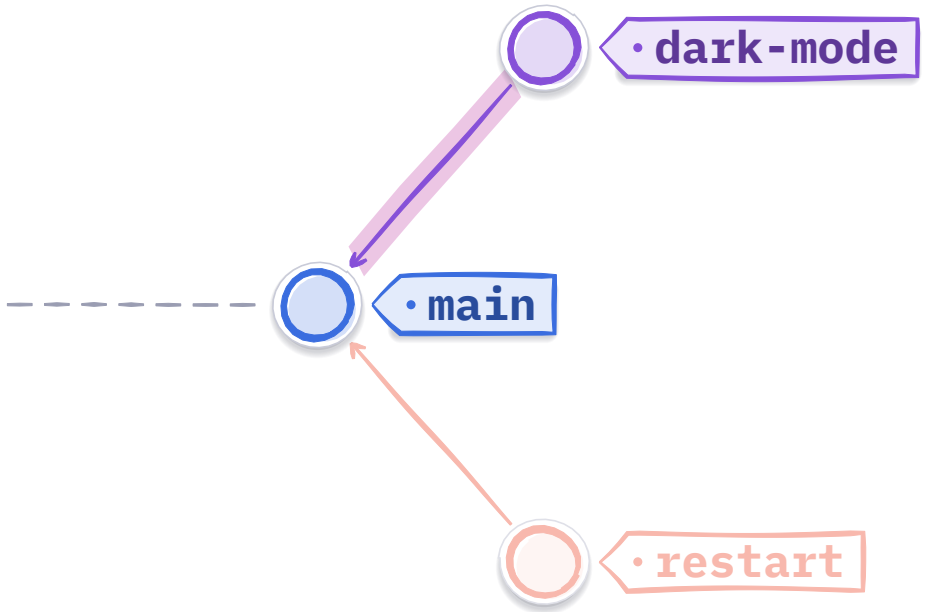
you typed by you

Merge dark mode into main.

A merge brings another branch's work into the branch we are on.

So it runs in `game/`, the folder on `main`.

Dark mode was built directly on `main`'s commit.



Merge dark mode into main

agent session ~/game

you typed by you

Merge dark mode into main.

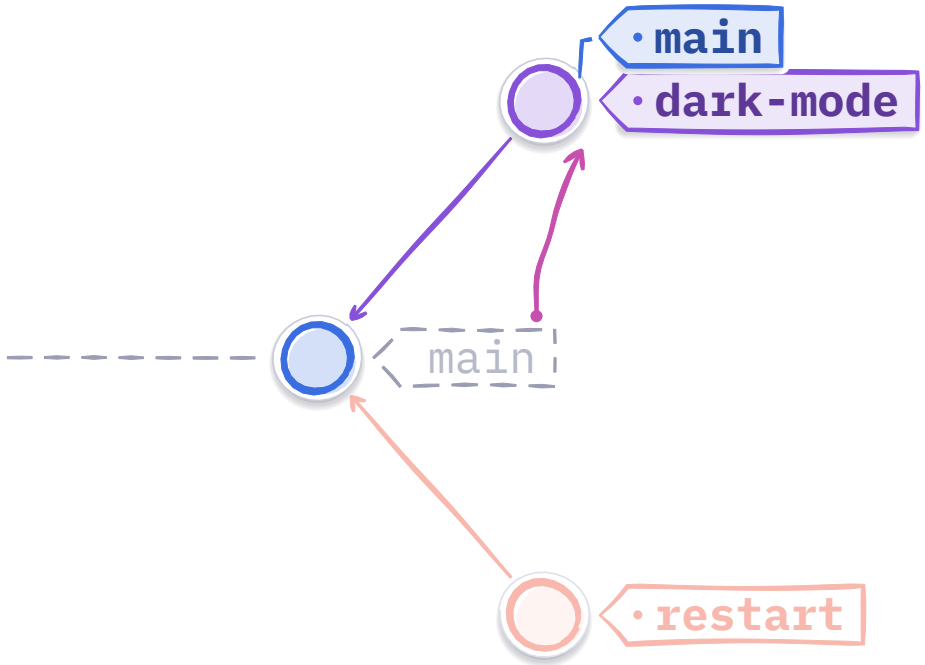
tool call asked by the model

Bash `git merge dark-mode`

tool result output trimmed

```
Updating 686e1c5..f119f79
Fast-forward
```

game/ now has dark mode.



Fast-forward: main moves to a commit that exists. No new commit.

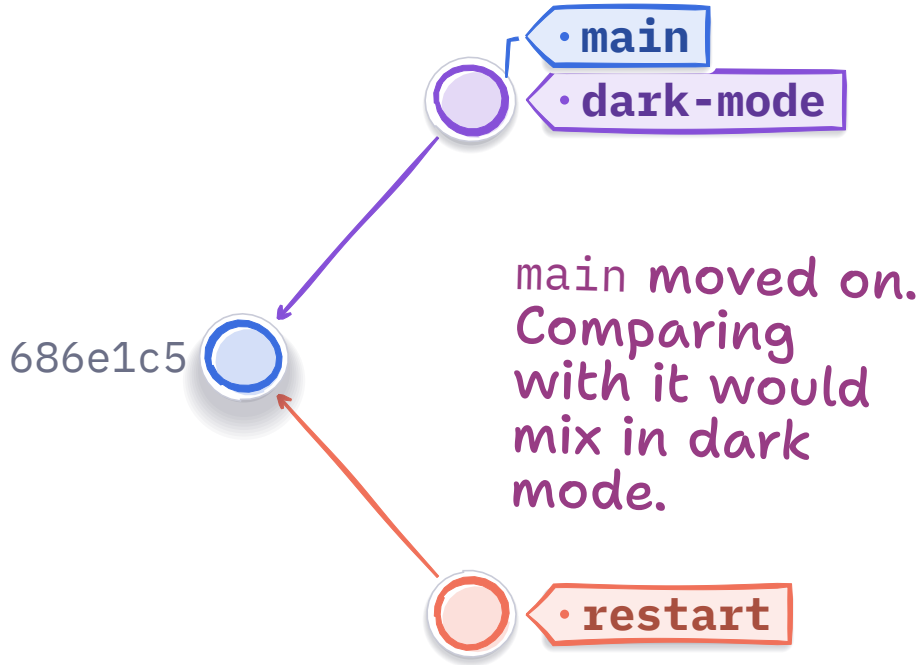
Review restart from where it started

shell

```
~/game-restart $ git diff 686e1c5
```

game.js — an excerpt of the diff

```
- const controlsHelp = "P: pause";
+ const controlsHelp = "P: pause | R: restart";
  ⋮
+ function restart() {
+   Object.assign(game, createGame());
+   paused = false;
+ }
+
+ keyActions.r = restart;
```



run in `game-restart/`

- R restarts the game
- P still pauses

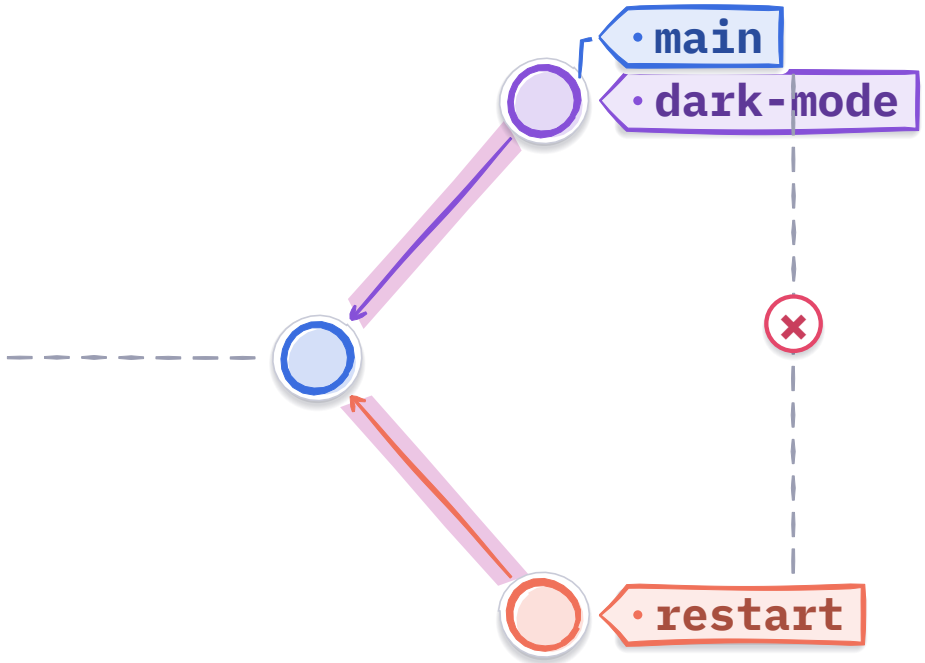
Merge restart: two lines of work

agent session ~/game

you typed by you

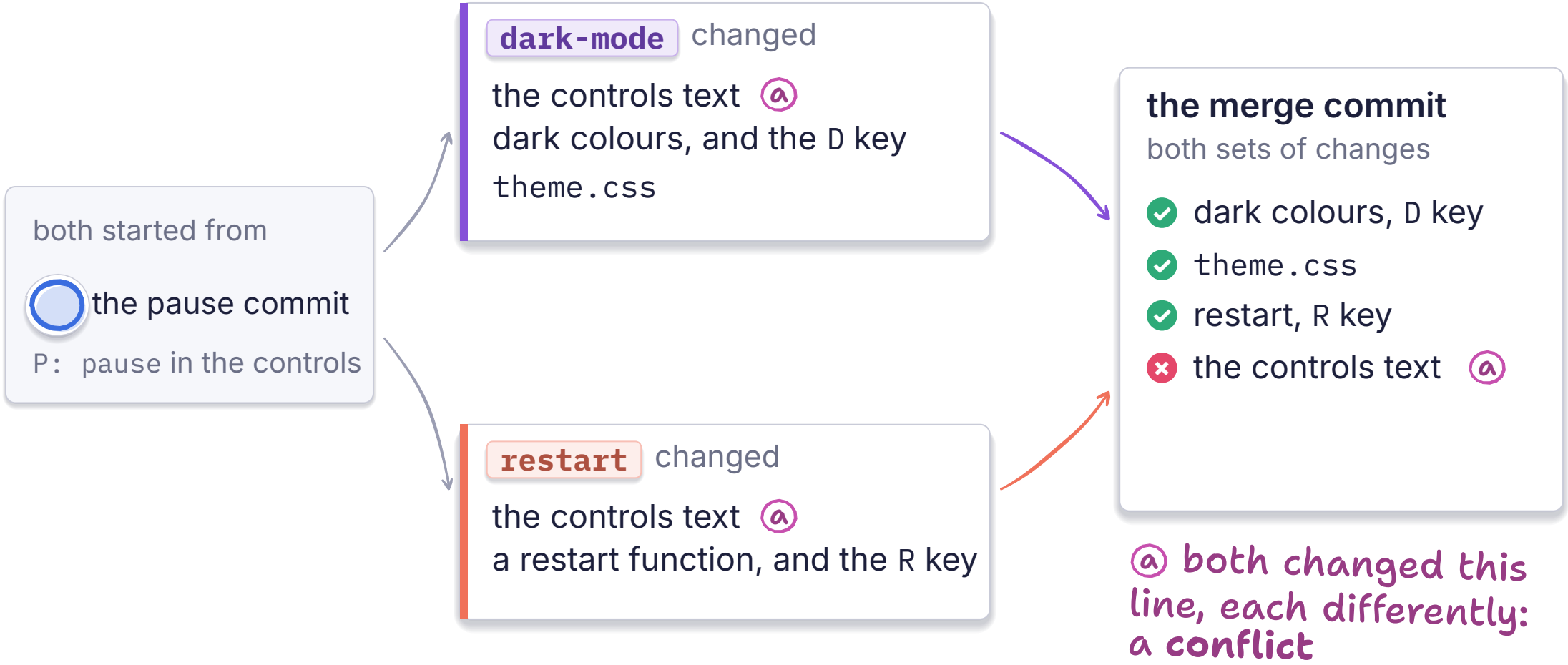
Merge restart into main.

Moving `main` to restart **would drop dark mode.** No fast-forward this time.



Each branch has new work the other lacks.

How git combines two lines of work



Merge restart: two lines of work

agent session ~/game

you typed by you

Merge restart into main.

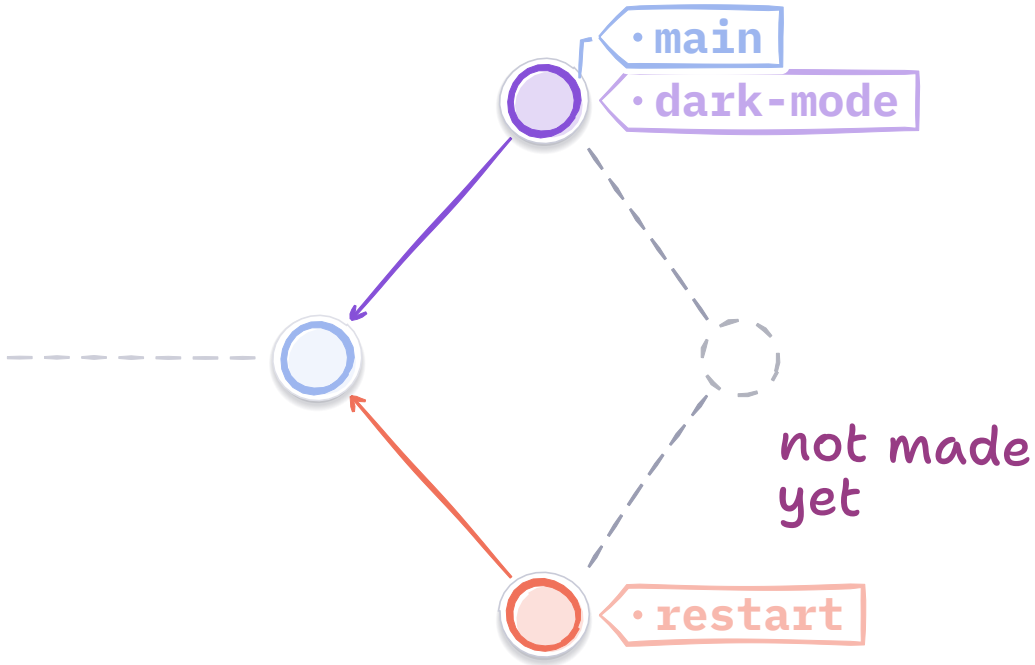
tool call asked by the model

```
Bash git merge restart
```

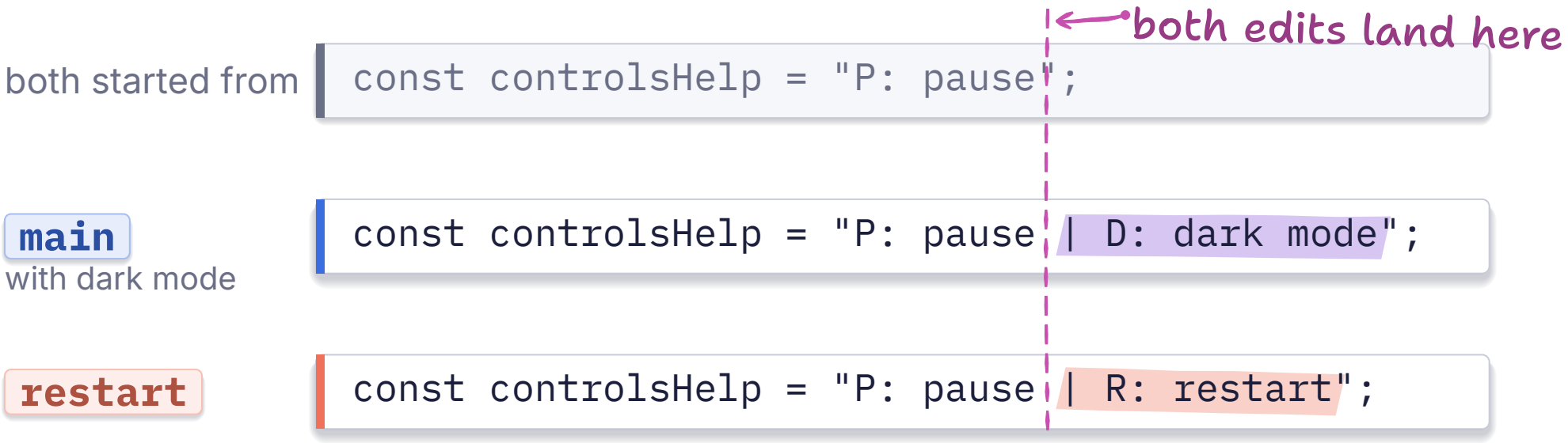
tool result run by the harness

```
Auto-merging game.js
CONFLICT (content): Merge conflict
in game.js
Automatic merge failed; fix conflicts
and then commit the result.
```

Git stops on one line both features changed.



Both features changed the same line



Git cannot decide which version is right. **We want both controls listed.**

Both features changed the same line

game.js — in game/, mid-merge

```
 7 const STEP = 6;
 8
 9 <<<<<<< HEAD
10 const controlsHelp = "P: pause | D: dark mode";
11 =======
12 const controlsHelp = "P: pause | R: restart";
13 >>>>>>> restart
14
15 const INVADER = [
```

Both features changed the same line

game.js — in game/, mid-merge

```
const STEP = 6;

<<<<<<< HEAD
const controlsHelp = "P: pause | D: dark mode";
=======
const controlsHelp = "P: pause | R: restart";
>>>>>>> restart

const INVADER = [
```

Both features changed the same line

game.js — in game/, mid-merge

```
 7 const STEP = 6;
 8
 9 <<<<<<< HEAD
10 const controlsHelp = "P: pause | D: dark mode";
11 =======
12 const controlsHelp = "P: pause | R: restart";
13 >>>>>>> restart
14
15 const INVADER = [
```

Alternatives to resolve, **not code to keep.**

Both features changed the same line

game.js — in game/, resolved

```
 7 const STEP = 6;
 8
 9 const controlsHelp = "P: pause | D: dark mode | R: restart";
10
11 const INVADER = [
```

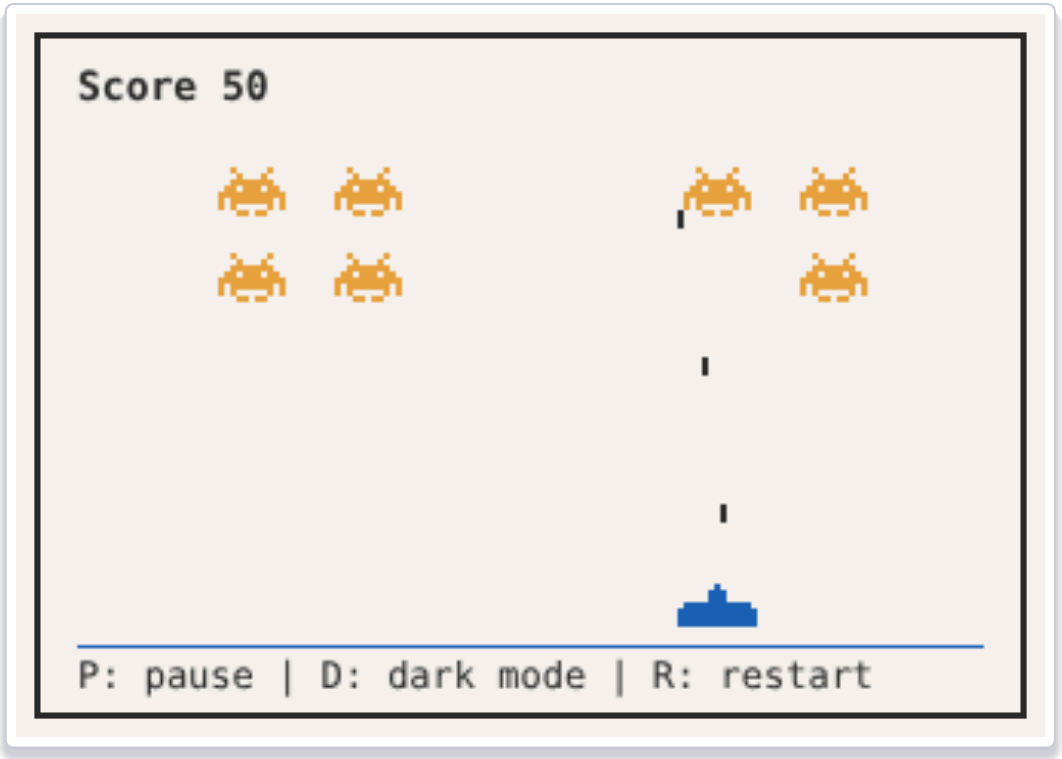
All three controls, and no markers left.

you typed by you

Keep both features: list all three controls.

Say what the result should do. Then read what the agent wrote.

Try all three controls in the combined game



screenshot · the merged game, run in `game/`

- **P** pauses and resumes
- **D** toggles the theme
- **R** restarts the game

Commit the merge: one commit, two parents

agent session ~/game

you typed by you

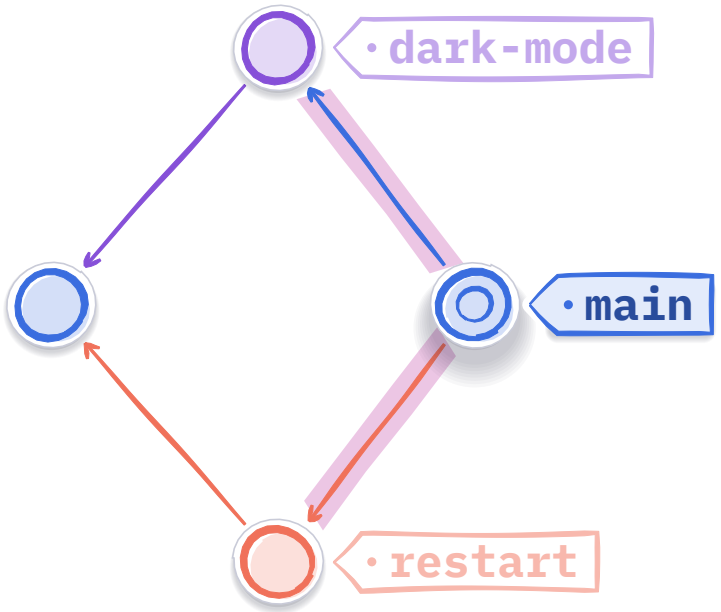
Finish the merge.

tool call asked by the model

Bash

```
git add game.js
git commit -m "Merge restart and dark mode"
```

`main` now has pause, dark mode and restart.



Two parents: the dark-mode commit and the restart commit.

Remove the finished folders and branches

agent session ~/game

you typed by you

Remove the finished folders and branches.

Check them for uncommitted work first.

tool call asked by the model

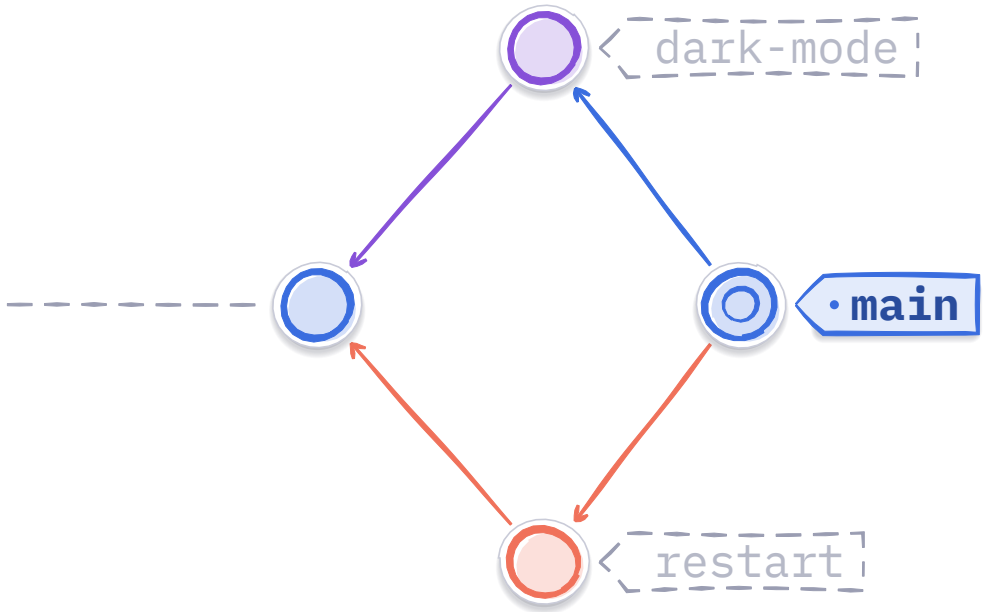
Bash

```
git worktree remove ../game-dark
git worktree remove ../game-restart
git branch -d dark-mode restart
```

tool result run by the harness

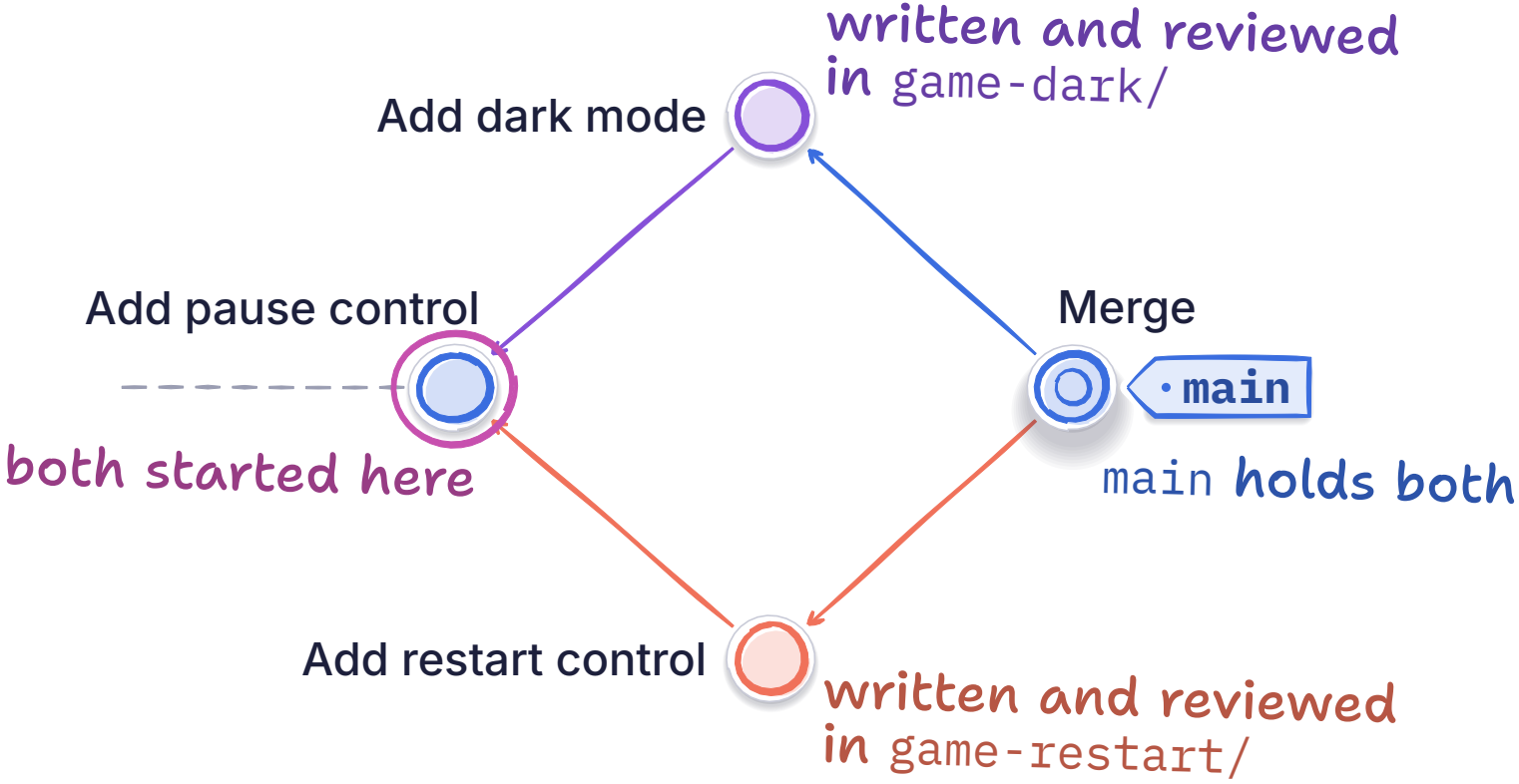
```
Deleted branch dark-mode (was f119f79).
Deleted branch restart (was d1a5c7a).
```

The names go. The merged commits stay in `main`'s history.



A feature we do not want is simply **never merged** into `main`.

Read the result from the history



The routine

1. Give each agent its own branch and folder.
2. Review its feature.
3. Merge, then check the result.

See you in the next video.